Healthcare staff routing problem using adaptive genetic algorithms with adaptive local search and immigrant scheme

Sinthamrongruk, Thepparit; Dahal, Keshav; Satiya, Oranut ; Vudhironarit, Thishnapha ; Yodmongkol, Pitipong

Published: 01/03/2017

Healthcare Staff Routing Problem using Adaptive Genetic Algorithms with Adaptive Local Search and Immigrant Scheme

Thepparit Sinthamrongruk[12], Keshav Dahal[1], Oranut Satiya[3], Thishnapha Vudhironarit[4], Pitipong Yodmongkol[2]

[1]University of the West of Scotland, UK
[2]Chiang Mai University, Thailand
[3]Phramongkutklao Hospital, Thailand
[4]Phramongkutklao College of Medicine, Thailand
Email: {thepparit.sinthamrongruk, keshav.dahal} @uws.ac.uk
{oranutsatiy, thishnapha.vudhironarit}@gmail.com
pitipong.y@cmu.ac.th

***Abstract*— Healthcare staff routing to provide healthcare service to the patients is one of the real-world scheduling problems similar to multiple travelling salesman problems (MTSP). Healthcare staff members provide daily medical services at patients' homes. The service provider authority has to schedule these staff in an effective and efficient way so that it achieves the minimum total cost. The aim of this study is to propose an Adaptive Local Search based on Genetic Algorithm (GA) to solve Healthcare Staff Routing Problem. Two new types of Adaptive Local Searches have been proposed to explore the optimal solutions. Also, Immigrant Scheme has been applied to improve the performance of the proposed GA. With this feature, we make an effort to motivate the GA to replace population occasionally by calling the best GA chromosome when the GA struggles at the local optimal solution. By the proposed algorithm, an effective routing schedule for staff members is generated. Our empirical study demonstrates that the proposed GA with Adaptive Local Search and Immigrant Scheme outperforms its rival methods in terms of the sum of distances.**

Keywords—adaptive local search, adaptive genetic algorithm; constructive scheduling; healthcare service;k-mean algorithm;

I. INTRODUCTION

Health care service plays a fundamental and essential role in daily life. The service starts with patient's requirement, which must be done by qualified staff members. The individual variety of qualifications comprises, for example, the number of languages spoken, or a license to administer drugs. The member staffs referred to as nurse or caregivers are typically equipped with private cars, bikes or use public transport going to patient' homes from the health care central office under specific time windows. Thus, an effective scheduling of the health care service is indispensable for the service. Basically, working schedule is generated as manual by planners for arrangement timetable referred to "Healthcare Staff Scheduling Problem (HSSP)". This problem also associates with a lot of resources such as staff members, preference and cost under specified time and capacity constraints.

HSSP has received intensive attention since the pioneering work in the UK since 1974 [1]. After that, there are many researchers who have investigated in HSSP. Several methods have been used for solving HSSP in different regions and problem domains such as routing and scheduling; major points of those researches tend to focus on only one specific perspective and avoid the other.

The purpose of this research is to solve HSSP using three-step scheduling technique by dividing the problem into different sub-problems and then finding solutions with these steps for route scheduling, resource selection, and local improvement. ***Routes scheduling*** focuses on how to arrange effective routes for staff with minimum distance travel time and travel cost. ***Resource selection*** points to match qualified staff to each route with the minimum cost and the preferences by the scheduler and also a customer representative under feasible time constraint. The ***Local Improvement*** enhances output solution generated by the resource selection using swapping task based on the cost function

In this paper, we focus on an improved scheduling algorithm for solving HSSP. The improved algorithm will be applicable for real life situations using combined approach based on Hybrid Evolutionary Algorithms to generate service routes. The Local Search techniques, Constructive scheduling, and Immigrant Scheme are then used to aid for solution improvement. The major objective of this research is to design an efficient schedule with optimizing cost functions under capacity constraints.

The paper is organized with following sections. A literature review is written in Section 2. Descriptions of HSSP including problem details, Constructive scheduling and adaptive GA are explained in Sections 3. Section 4 provides the result of all scenarios, followed by the computational time of experiments in Section 4. Finally, the conclusion and the direction for future works are presented in Section 5.

II. Literature Reviews

A. Health Care Service and problem solving

Several techniques have been proposed to solve HSSP such as mathematics, heuristics or even meta-heuristics. As part of mathematical programming methods, Felici and Gentile [2] presented an integer programming model that maximizes the total satisfaction of the nursing staff. Also, Bard and Purnomo [3] adopted the column generation scheme to solve the problem in terms of minimising the nursing staff members' violating preferences. Next, fuzzy theory was applied to a multi-objective integer programming model in order to determine the changeable factors that influence nurse satisfaction [4]. Bredström and Rönnqvist [5] proposed a mathematical programming model combined with OPT, a Local Search to handle vehicle routing and nurse scheduling problem under time windows and additional temporal constraints. Experimental results illustrated a positive effect. Constantino et al. [6] proposed a new deterministic heuristic algorithm to solve a nurse scheduling problem consisting of two phases; a constructive scheduling phase and an improving phase.

Evolutionary algorithm, a population-based approach has recently become more popular in nurse schedule management since 2000. Cai et al.[7] applied a Genetic Algorithm to minimize the costs of allocating staff for over-time work, as well as to optimize a solution for scheduling staff of mixed skills under multi-criteria. In addition, Harmony search, a metaheuristic algorithm, was used for the nurse scheduling problem in a hospital in Malaysia [8]. Alternatively, Todorovic and Petrovic [9] dealt with the difficulty of nurse scheduling using a bee colony optimization which is able to eliminate some ineffective plans from the neighborhood solution. Particle-swarm-based approach (PSO) was applied for HSSP in 2007. Akjiratikarl et al. introduced PSO and a heuristic assignment scheme to generate the schedule of caregivers in the UK [10]. In 2013, Gao and Lin [11] presented research concerning the nurse scheduling problem. The goal of this research was to maximize the happiness level of caregivers working under hospital regulation using classical PSO to solve a mathematical model compared with manual scheduling. To improve traditional metaheuristic methods, an electromagnetic algorithm, combining three Local Search methods for variable neighbourhood, was applied to solve the nurse scheduling problem by Maenhout and Vanhoucke [12]. An immigrant scheme, a local improvement, was developed to achieve a better result in comparison with the traditional Genetic Algorithm. The immigrant scheme is designed to recall the best solution or population stored in memory when the new return output is lowered continually [13]. A recovery scheme for the Genetic Algorithm is added to tackle more complex problem for HSSP. Chang-Chun Tsai et al.[14] proposed a new two-stage model for HSSP. The purpose of this method is to reduce infeasible solutions in GA designed specifically for our nurse preference scheduling. Experimental results showed that it can repair the infeasible solutions. To solve larger HSSP, some researchers proposed a hierarchy problem solving procedure to divide the problem into smaller sub-problems in diverse domains in order to extend the search space and find the global optimal solution [15].

III. Problem Details

In this section, HSSP is explained step by step and also the new modified route scheduling approach using Constructive Scheduling is described as follows:

A. Problem description

The problem description given below is derived and adopted from a referenced paper [10] for HSSP.

- Staff members are recommended to service only up to 5 patients every working day as higher number of patients may lead to fatigue and exertion on the staff members.
- Staff members or caregivers are assigned jobs according to patient requirement.
- Each job has the same priority. Patients cannot demand specified service time like e.g. 8.00 a.m. or 1.00 p.m. because it is free service from local council.
- Each care worker starts from Home care office at 8.00 after finishing their assigned jobs and each job must be completed within assigned time windows. Also, Operating time has been investigated in the research. Default value is an hour per service. Thus, each route contains approximately 5 tasks each day.
- The location of patient homes is defined by Geolocation (Latitude and Longitude coordination).
- The travel speed depended on modes of transportation for care workers comprising of public transportation with taking train, subway and public buses, and walking.
- In this paper, the service is assumed as physiological service

B. Proposed techniques: Route scheduling

The goal of route scheduling is to create routes for staff members to service patients at different task locations with the total shortest path. Genetic Algorithm (GA) is one of the most popular searching algorithms as it is acceptable as global optimization technique for complex problem without applying excessive complex mathematical model. GA processes a structure using chromosome representation and randomized operators to evolve solution. Problems are encrypted as chromosome which represents possible solution before using crossover and mutation operator to create new offspring.

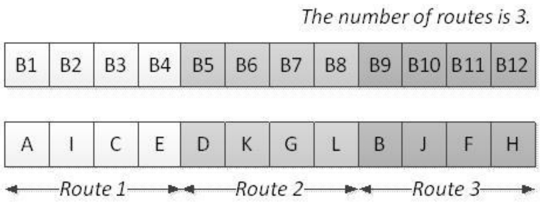

Fig. 1. bits MTSP chromosome representation with n-route = 3.

In this work, the multiple traveling salesman problems (MTSP) chromosome representation is designed to develop for experiments. In HSSP, caregivers offer daily medical services at patient's homes. All member staffs have to start their work from the Health Care Office referring to MTSP. Service schedule provides multiple routes or salesmen within a single chromosome which complies with HSSP. The number of salesmen is defined as the number of routes collected to one chromosome representation. Fig. 1 and 2 are an example that each chromosome is designed to contain three routes(n-route =3) where Route-1 starts from the Health Care Office going to $Task_A$, $Task_B$, $Task_C$, $Task_D$, and then go back to the office at the end. Route-2 begins from the office going to $Task_D$, $Task_k$, $Task_G$, $Task_L$ and going back to the office like the first route. The last one is route-3 which starts at the same Health Care Office and going to $Task_B$, $Task_J$, $Task_F$, $Task_H$, and then going back to the home office.

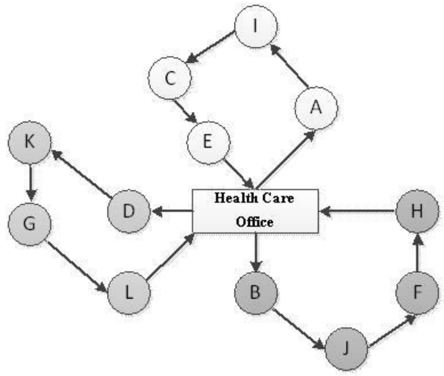

Fig. 2. MTSP chromosome representation with n-route = 3.

C. Genetic Algorithms

The GA procedure is described as follows:

1) Initial population generation

In this research, Constructive Scheduling (CS) with K-mean, a clustering algorithm [84] is used to generate initial solution for solving HSSP. This generation technique partitions the locations of patient homes into k sub-groups or k-routes where K is the number of clusters. The quality of clustering technique bases on the centroid values which combine closer task location to its route (see equation (1)).

$$J\,(C) = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \qquad (1)$$

2) GA operator: crossover and mutation

Two operators including crossover (PMX) and mutation are used to generate new offspring. The crossover rate (*Pc*) is defined as the ratio of the number of genes that will be crossover to the population size which a range between 0% up to 100% while the mutation operator flips or alters one or more bit values randomly in a chromosome. The mutation rate P_m is defined as the ratio of the number of genes that will be mutated to the population size ranging from 0% up to 100%. Both these operators also assist the offspring exploration for searching the global optimal solution.

3) Parent selection

Parent selection is a procedure to pick up some populations adding to parent chromosome members. The possibilities depending upon the fitness values (minimizing problem-lower is better) are added at the last extensional bit of whole target populations before using roulette wheel for selection. Fig.3 is an example with four populations. The first population is 75 fitness value-units translated to 23.53% possibilities on the roulette wheel. The remainders are 17.65%, 35.29%, and 23.53% respectively. The sum of possibilities in the roulette wheel is 100%.

4) Adaptive GA

There are two essential parameters which are controlled in traditional GA: crossover rate P_c and mutation rate P_m. Technically these parameters are set as constant values before running experiment. These numbers affect significantly the quality of reproduction conducing to a variety of computational time. This method allows GA to adjust its parameters during running GA automatically. Feedback from the fitness value is used to control *Pc* and adaptive *Pm* based on equation (2).

$$P_{x(new)} = \begin{cases} \frac{P_{x(old)}+1}{2}, & fitness_{current} > fitness_{old} \\ P_{x(old)}, & fitness_{current} \le fitness_{old} \end{cases} \qquad (2)$$

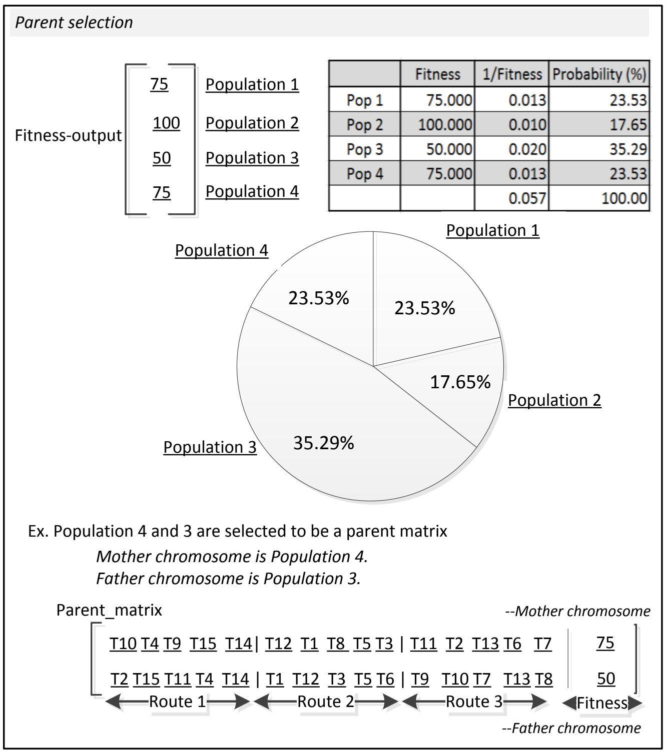

Fig. 3. MTSP chromosome representation with n-route = 3.

D. Adaptive Local Search

1) Original Local Search (LS) or 2-OPT

In the HSSP, 2-OPT is defined as Local Search included in our GA to enhances results of GA. Original 2-OPT algorithm begins by removing two edges such as job_i - job_j and job_k - job_l edges from the tour in a route, and reconnecting the two paths from job_i - job_k and job_j - job_l. In Fig.4 is an example of six patient locations from A to F represented as a graph and each edge between two points in terms of distance-unit. For example, *AF* edge is 2 units and *FE* edge is 24 units respectively. The sum of original route is 47 units. Edge DC and EF are selected as random for 2-OPT. If DC and EF are not adjacent edge, we can replace DC and EF with DE and CF. To confirm the replaceable solution, the total distance of new route is recalculated with 41 units which is lower than the original one at 47 units.

2) Adaptive 2-OPT

Due to its ease of use, 2-OPT has been advantageous for Meta-heuristics. However, 2-OPT takes a huge computational time due to the fact that as the number of repeating sub-loops is fixed all the time of execution. To handle the problem, we proposed Adaptive Local Search inspired from Adaptive GA. With this new improvement, the number of repeating loop of Local Search adjusts automatically depending on current GA outputs of iterations. Two formulations of Adaptive Local Searches including type1 (ADL1) and type2-Adaptive Local Search (ADL2) are proposed in this research. The latter is an improved procedure where the replicable rate can be adjusted both sides of previous rate (higher and lower sides) (see equation (3)) while the aim of the former is to decrease the rate of 2-OPT slightly depending equation (4).

$\boldsymbol{Type1 - Adaptive\ Local\ Search\ (ADLS1)}$

$$P_{2opt(new)} = \begin{cases} P_{2opt(old)}++, & fitness_{current} > fitness_{old} \\ P_{2opt(old)}--, & fitness_{current} \leq fitness_{old} \end{cases} \quad (3)$$

$\boldsymbol{Type2 - Adaptive\ Local\ Search\ (ADLS2)}$

$$P_{2opt(new)} = \begin{cases} \frac{P_{2opt(old)}+1}{2}, & fitness_{current} > fitness_{old} \\ P_{2opt(old)}, & fitness_{current} \leq fitness_{old} \end{cases} \quad (4)$$

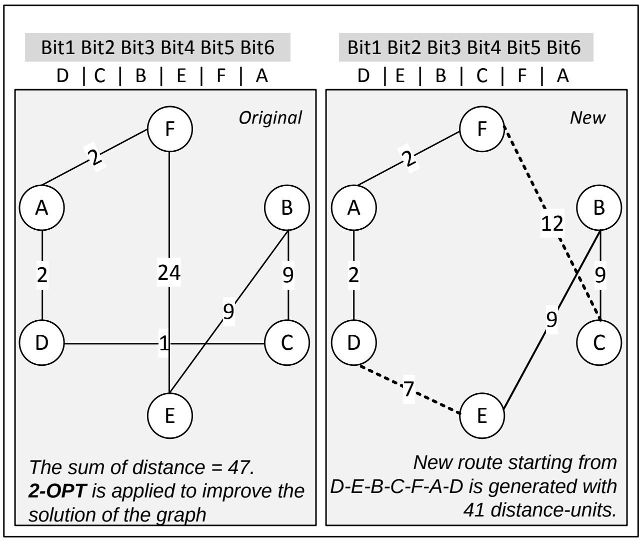

Fig. 4. Original route and new route with 6 patien locations

E. Immigrant Scheme(IS)

IS has been applied to improve the performance of the algorithms. In this procedure, we make an effort to stimulate the GA when it struggles at the local optimal solution by calling the best chromosome of GA to replace in population tray. The output vector of GA has been motivated continually going to the optimal solution. This feature is activated every third generations if returned outputs are on degenerate side.

F. Dataset

Dataset is defined as six inputs including task number, x and y coordinates, operating time, staff requirement and type of locations (patient location and the home office location). Euclidean distance is implemented according to the paper [3] that used two dimensional symmetric problems.

G. Design of experiment

Ten parameters are used in all experiments to create scenarios. Each scenario repeats three times to find average fitness value and average computational time (see **Table I**). The number of simulated patient locations is referred from Duque et al.[16]. Their study used data locations between 26 to 109.

TABLE I. PARAMETER-SETTING

Parameters	***Level of parameters***	
Crossover rate (Pc)	*0-20%*	
Mutation rate (Pm)	*0-20%*	
Constructive Scheduling (CS)	*Enable*	
Adaptive GA (ADGA)	*Enable*	*Disable*
Adaptive GA with Immigrant Scheme (ADGA+IM)	*Enable*	*Disable*
Local Search (LS)	*Enable (fixed rate)*	*Disable*
Adaptive Local Search (ADLS)	*ADLS1 and ADLS2 (0-20 % of P_{2OPT})*	
Number of patient locations	*40,60, and 100 patient locations*	
Maximum iteration	*300*	
Replication	*3*	

IV. EXPERIMENTAL RESULTS

The proposed algorithms have been implemented in R language on laptop Intel i5 systems with 8 GB RAM, running at 1.7 GHz. All of scenarios are generated from *Pc*, *Pm*, P_{2OPT}. Simultaneously, Adaptive GA with CS, IS and Adaptive Local Search are utilized to create major scenarios as follows:

1) Adaptive GA *(ADGA)*

2) Adaptive GA with Immigrant Scheme *(ADGA+IM)*

3) Adaptive GA with Immigrant Scheme and Fixed Local Search *(Fixed-LS)*

4) Adaptive GA with Immigrant Scheme and Adaptive Local Search based on $equation_1$ (ADLS1)

4) Adaptive GA with Immigrant Scheme and Adaptive Local Search based on $equation_2$ (ADLS2)

Dominant experimental results are shown in terms of fitness value and computational time (see in Table1 and 2). The first value is the sum of total distance of all routes. The constraints are used in a way that the staff must visit each task location and each route contains five locations.

Table II provides a comparison between Adaptive GA and Adaptive GA with immigrant scheme. Three generated datasets including 40, 60, and 100 data locations are used for the experiment

For 40 patient locations, ADGA+IM shows better solution compared with ADGA in terms of the fitness value for all scenarios at 139.94, 141.68, 139.41, and 141.31 respectively while the average computational time of both methods is a small difference at 8.7%. For dataset with 60 and 100 patient locations, ADGA+IM is still more effective than traditional methods. The average fitness value compared to ADGA shows on positive side at 254.46 and 421.72 distance units. Thus, ADGA+IM is more effective than tradition Adaptive GA and it is selected to be the standard GA method for Local Search experiment.

In **Table III**, *Fixed-LS* is defined as the standard GA with LS. The constant number of replaceable edge between 5 to 20% is used for data exploration while *Pc* and *Pm* are set at 5% and 20% respectively. *ADLS1* and *ADLS2* are Adaptive GA with Adaptive Local Search. The starting percentage of P_{2OT} is passed to be initial local search rate (*intLSR*) before GA execution. During running GA, *intLSR* is adjusted automatically according to (3) and (4) respectively.

As part of Local Search implementation, the results of Fixed-LS outperform the standard GA in terms of the average fitness value and computational time. For dataset with 40 patient locations, the average outcome improves from 140.58 units to 129.3 units or 7.61%. In contrast, the average computational times rise remarkably from 5.37 to 14.25 seconds. For 60 and 100 locations, results continue the same trend. The average fitness values of two data locations enhance from 244.46 to 236.4 and 422.21 to 394 fitness distance units or 3.3% and 6.68% respectively while the computational time increases almost 3 times and 2.5 times as compared to 40 data locations. In addition, other parameters including *Pc*, *Pm*, and *intLSR* affect results meaninglessly. Another interesting aspect is significantly *IntLSR* influencing on the computation time. *IntLSR* at 5% conduces to lower computational time for whole datasets.

From data mentioned above, even though Fixed-LS can help standard GA getting better solution, it consumes a big number of executional times. Thus, ADLS1 and ADLS2 are proposed to handle this issue. Unexpectedly, ADLS1 shows outstanding results compared its rival methods. For 40 and 60 data locations at scnarios_{20} (*Pc*=5%, *Pm*=20%, and *intLSR*=10%), it shows the improved reproductions of 121.77 and 215.92 distance-units respectively. In contrast, it takes too much computational time at 21.38 and 41.14 seconds. For the last dataset with parameter-setting with *Pc*=5%, *Pm*=5%, and *intLSR*=5%, proposed technique provides the best solution. On the other hand, ASLS2 provides less effectiveness than the first one in terms of the fitness value; it shows improved overall computational time. The average computational times are 7.659, 9.03, and 16.97 respectively.

V. CONCLUSION

Compared to the original, *ADGA+IM* shows outstanding results for three simulated datasets including 40, 60, and 100 patient locations in terms of cost and executional time. There are two major reasons to support ADGA+IM providing outstanding solutions. Firstly, calling the best population for population creation reduces CPU time because CPU does not to spend time to create new population.

Local Search takes higher computational time for all experiment. To handle with this problem, we proposed Adaptive Local Searches: ADLS1 and ADLS2.

The solutions gained in the experiment indicate that ADLS1 is the best type of GA compared to its rival algorithms in terms of the fitness value. In contrast, this method consumes highest time. For ADLS2, it gives results improved slightly while the computational time increases meaninglessly. For entire experiments, *Pc* and *Pm* barely affect the mode. When *Pc and Pm* are set in low level, they can provide better solutions with less computational time.

TABLE II A COMPASION OF ADAPTIVE GA AND ADAPTIVE GA WITH IMMIGRANT SHCEME

Scenario	Parameter-setting			40 Locations				60 locations				100 locations			
	pc	**pm**	**Type of GA**	**Fitness**	**Avg.**	**Time**	**Avg.**	**Fitness**	**Avg.**	**Time**	**Avg.**	**Fitness**	**Avg.**	**Time**	**Avg.**
1	5	5	ADGA	141.86	143.43	4.93	4.94	251.27	259.31	7.68	7.57	422.67	422.21	14.31	15.84
2	5	20	ADGA	144.92		4.84		259.47		7.32		418.67		17.64	
3	20	5	ADGA	142.11		5.00		260.21		7.60		423.41		15.04	
4	20	20	ADGA	144.82		4.99		266.28		7.69		424.08		16.37	
5	5	5	ADGA+IM	139.94	140.58	5.36	5.37	253.32	254.46	8.84	8.82	420.48	421.72	17.36	16.87
6	5	20	ADGA+IM	141.68		5.43		255.00		8.37		417.91		15.34	
7	20	5	ADGA+IM	139.41		5.44		255.16		8.93		425.55		16.80	
8	20	20	ADGA+IM	141.31		5.24		254.36		9.12		422.94		18.00	

TABLE III REMARKABLE RESULTS OF FOUR ADAPTIVE GENETIC ALGORITHMS AND ADAPTIVE LOCAL SEARCHES

Scenario	Parameter-setting				40 Locations				60 locations				100 locations			
	pc	pm	Type-LS	intLSR	Fitness	Avg.	Time	Avg.	Fitness	Avg.	Time	Avg.	Fitness	Avg.	Time	Avg.
1	5	5	Standard	0	139.94		*5.36		253.32		8.84		422.67		*17.36	
2	5	20	Standard	0	141.68	140.58	5.43	5.37	*255	254.46	*8.37	8.82	*418.67	422.21	15.34	16.87
3	20	5	Standard	0	*139.41		5.44		255.16		8.93		423.41		16.8	
4	20	20	Standard	0	141.31		5.24		254.36		9.12		424.08		18	
5	5	5	Fixed-LS	5	131.2		9.5		243.91		14.4		404.22		25.49	
6	5	20	Fixed-LS	5	132.42		9.09		233.6		*14.23		399.37		*25.26	
10	5	20	Fixed-LS	20	*121.78	129.3	20.44	14.25	232.14	236.4	31.14	22.41	386.89	394	54.76	38.68
11	20	5	Fixed-LS	5	134.24		*9.09		239.83		14.3		406.17		26.26	
15	20	5	Fixed-LS	20	122.99		20.61		*227.89		33.32		*376.87		54.75	
17	5	5	ADLS1	5	122.74		*21.38		221.15		*41.14		355.67		90.58	
18	5	20	ADLS1	5	124.48		23.53		217.49		50.82		360.79		*85.44	
19	5	5	ADLS1	10	122.45	124.1	26.88	26.35	216.41	219.6	48.91	50.08	361.27	359.2	114.53	100.01
20	5	20	ADLS1	10	*121.77		26.99		*215.92		51.05		355.77		89.43	
23	20	5	ADLS1	5	129.89		22.7		220.95		46.13		*345.38		91.39	
30	5	20	ADLS2	5	136.23		7.65		258.16		8.99		416.26		*16.48	
33	5	5	ADLS2	20	139.48		7.48		253.02		*8.54		418.55		16.75	
37	20	5	ADLS2	10	134.49	136.5	*7.4	7.659	258.48	255.6	9.03	9.03	420.58	420.2	17.35	16.97
38	20	20	ADLS2	10	137.82		7.63		253.43		9.27		*407.37		17.29	
39	20	5	ADLS2	20	*133.88		7.88		*250.13		9.25		426.91		17.17	

* ADL1: type 1-Adaptiie Local Search and ADL2: type 1-Adaptiie Local Search

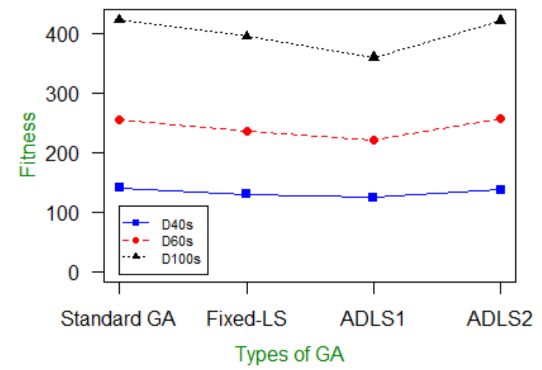

Fig. 5. Fitness values

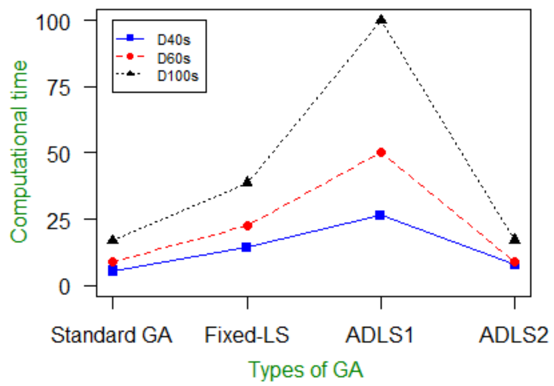

Fig. 6. Fitness values

REFERENCES

[1] G. G. A. H. A. C. L. Aurora Fernandez, "A Model for Community Nursing in a Rural County," *Operational Research Quarterly (1970-1977)*, vol. 25, no. 2, pp. 231-239, 1974.

[2] G. Felici and C. Gentile, "A Polyhedral Approach for the Staff Rostering Problem," *Management Science*, vol. 50, no. 3, pp. 381-393, 2004.

[3] J. F. Bard and H. W. Purnomo, "Preference scheduling for nurses using column generation," *European Journal of Operational Research*, vol. 164, no. 2, pp. 510-534, 7/16/ 2005.

[4] S. Topaloglu and H. Selim, "Nurse scheduling using fuzzy modeling approach," *Fuzzy Sets and Systems*, vol. 161, no. 11, pp. 1543-1563, 2010/06/01 2010.

[5] D. Bredström and M. Rönnqvist, "Combined vehicle routing and scheduling with temporal precedence and synchronization constraints," *European Journal of Operational Research*, vol. 191, no. 1, pp. 19-31, 2008.

[6] A. A. Constantino, D. Landa-Silva, E. L. de Melo, C. F. X. de Mendonça, D. B. Rizzato, and W. Romão, "A heuristic algorithm based on multi-assignment procedures for nurse scheduling," *Annals of Operations Research*, vol. 218, no. 1, pp. 165-183, 2014// 2014.

[7] X. Cai and K. N. Li, "A Genetic Algorithm for scheduling staff of mixed skills under multi-criteria," *European Journal of Operational Research*, vol. 125, no. 2, pp. 359-369, 9/1/ 2000.

[8] M. Hadwan, M. Ayob, N. R. Sabar, and R. Qu, "A harmony search algorithm for nurse rostering problems," *Information Sciences*, vol. 233, pp. 126-140, 6/1/ 2013.

[9] N. Todorovic and S. Petrovic, "Bee Colony Optimization Algorithm for Nurse Rostering," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 2, pp. 467-473, 2013.

[10] C. Akjiratikarl, P. Yenradee, and P. R. Drake, "PSO-based algorithm for home care worker scheduling in the UK," *Computers & Industrial Engineering*, vol. 53, no. 4, pp. 559-583, 2007.

[11] S.-C. Gao and C.-W. Lin, "Particle Swarm Optimization Based Nurses' Shift Scheduling," in *Proceedings of the Institute of Industrial Engineers Asian Conference 2013*, Y.-K. Lin, Y.-C. Tsao, and S.-W. Lin, Eds. Singapore: Springer Singapore, 2013, pp. 775-782.

[12] B. Maenhout and M. Vanhoucke, "An Artificial Immune System Based Approach for Solving the Nurse Re-rostering Problem," in *Evolutionary Computation in Combinatorial Optimization: 13th European Conference, EvoCOP 2013, Vienna, Austria, April 3-5, 2013. Proceedings*, M. Middendorf and C. Blum, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 97-108.

[13] C.-C. Lin, J.-R. Kang, D.-J. Chiang, and C.-L. Chen, "Nurse Scheduling with Joint Normalized Shift and Day-off Preference Satisfaction Using a Genetic Algorithm with Immigrant Scheme," *Int. J. Distrib. Sen. Netw.*, vol. 2015, pp. 9:9-9:9, 2015.

[14] C.-C. Lin, J.-R. Kang, and T.-H. Hsu, "A Memetic Algorithm with Recovery Scheme for Nurse Preference Scheduling," *Journal of Industrial and Production Engineering*, vol. 32, no. 2, pp. 83-95, 2015/02/17 2015.

[15] S. Yalçındağ, A. Matta, E. Şahin, and J. G. Shanthikumar, "The patient assignment problem in home health care: using a data-driven method to estimate the travel times of care givers," *Flexible Services and Manufacturing Journal*, vol. 28, no. 1, pp. 304-335, 2016.

[16] P. A. M. Duque, M. Castro, K. Sörensen, and P. Goos, "Home care service planning. The case of Landelijke Thuiszorg," *European Journal of Operational Research*, vol. 243, no. 1, pp. 292-301, 2015.